Supplementary Material of Click-Gaussian: Interactive Segmentation to Any 3D Gaussians

Seokhun Choi[1*], Hyeonseop Song[1*],
Jaechul Kim[1], Taehyeong Kim[2†], and Hoseok Do[1†]

[1] AI Lab, CTO Division, LG Electronics, Republic of Korea
[2] Dept. of Biosystems Engineering, Seoul National University, Republic of Korea

A GUI-based Implementation for Click-Gaussian

To showcase interactive segmentation and manipulation using Click-Gaussian, we design a Graphical User Interface (GUI) tool based on DearPyGui [2, 8], a fast and powerful GUI toolkit for Python. As shown in Fig. 1, our GUI is designed to allow users to easily click and segment objects at coarse and fine levels, and provides tools for real-time manipulation tasks such as resizing, translation, removal, and text-based editing for intuitive interaction with the segmented objects. The supplementary video demonstrates the effectiveness of our method in enabling real-time interactive scene manipulation, showcasing its fast and precise 3D segmentation performance. We encourage readers to view this video for a comprehensive understanding of the proposed approach's capabilities.

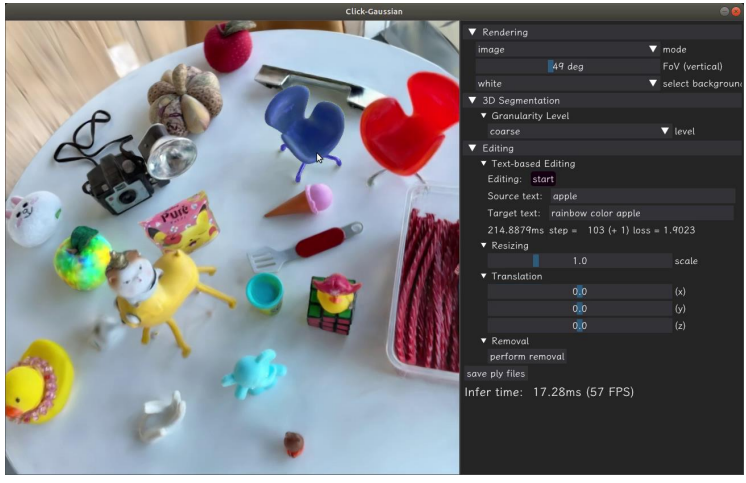

Fig. 1: Graphical User Interface (GUI) for Click-Gaussian.

[*] Equal contribution.
[†] Co-corresponding author.

B SAM-based Multi-level Mask Generation

We utilized the official code's *automatic mask generation module* for SAM mask creation, which extracts masks without distinguishing levels, allowing us to get only the highest-confidence segments in an image. These segments are then assigned to two masks by area: if multiple segments are assigned to a single pixel, the coarse-level mask prioritizes the identity of the larger segment, while the fine-level mask favors the identity of the smaller segments. This approach enables us to assign a single mask identity per pixel at each level, facilitating stable contrastive learning.

Comparative Analysis of Multi-level Mask Strategies. Our method can adopt SAM's three-level masks (whole, part, and subpart) in two ways: three-level-score and three-level-area. Each approach prioritizes the highest score segment and smallest segment, respectively, for each level. In these cases, we split $\mathbf{f}_i \in \mathbb{R}^{24}$ into three levels of granularity. As shown in Fig. 2, the three-level-area outperforms the three-level-score in fine-level mIoU due to finer-grained mask supervision (*e.g.*, egg white and yolk), demonstrating the efficacy of the area-based prioritization. Additionally, our method using two-level masks surpasses the three-level-area thanks to the mask completeness and training efficiency: It has fewer unassigned identities than the three-level-area and learns feature fields more efficiently with the same feature dimension of 24. For these advantages, we adopt the two-level granularity assumption.

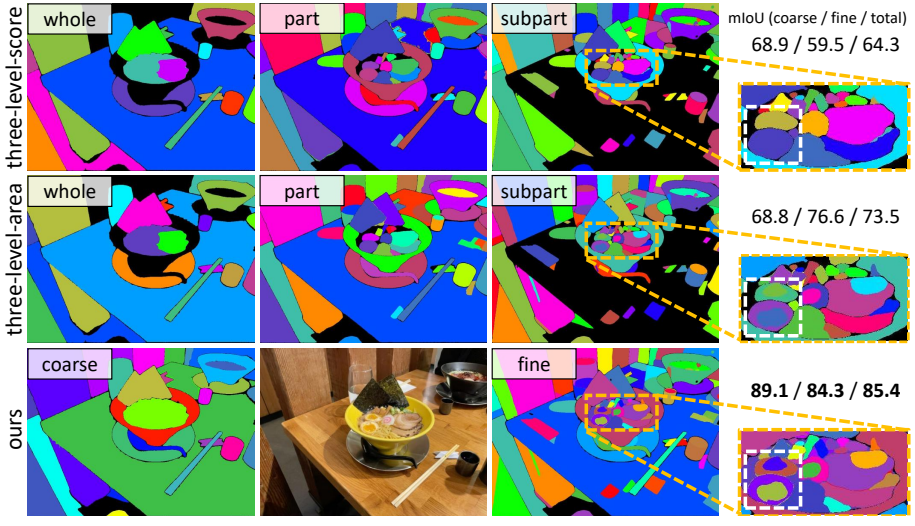

Fig. 2: Performance comparison of different mask generation strategies. Black areas indicate pixels with unassigned identities.

C Annotations for Evaluating Fine-grained Segmentation

We evaluate our approach's segmentation performance using the LERF-Mask dataset [9], a public real-world dataset for 3D segmentation tasks. This dataset comprises three scenes (Figurines, Ramen, and Teatime) [3] with manually annotated ground truth masks for semantically large objects, as shown in the first two rows of Fig. 3. To assess fine-grained segmentation performance, we additionally annotated masks for smaller objects within each scene using Make-Sense [7], a free online image labeling tool, as shown in the last two rows of Fig. 3. This additional annotation is necessary due to the lack of datasets suitable for fine-level comparison. Note that the annotation process was conducted independently from our segmentation experiments.

Fig. 3: Annotations for evaluating fine-grained segmentation. The first two rows of each scene show the ground truth annotations for evaluating coarse-level segmentation with two sampled test views. On the other hand, each scene's last two rows show the ground truth annotations for evaluating fine-level segmentation.

D Additional Experiments and Results

D.1 3D Editing in AI-generated Videos

OpenAI recently announced Sora [1], a groundbreaking text-to-video generation model, showing a promising path towards building general-purpose world simulators. These simulators can be further improved by enabling interactive modification of generated realistic environments through accurate and fast 3D segmentation methods like Click-Gaussian, enhancing their functionality and user interaction capabilities. To demonstrate Click-Gaussian's versatility in scene segmentation and manipulation on these generated scenes, we applied our method to videos (Santorini[1] and Snow-village[2]) generated by Sora. As shown in Fig. 4, after pre-training 3DGS on each generated video using COLMAP [5, 6], users can flexibly make desired modifications, resulting in more creative and diverse 3D environments with Click-Gaussian.

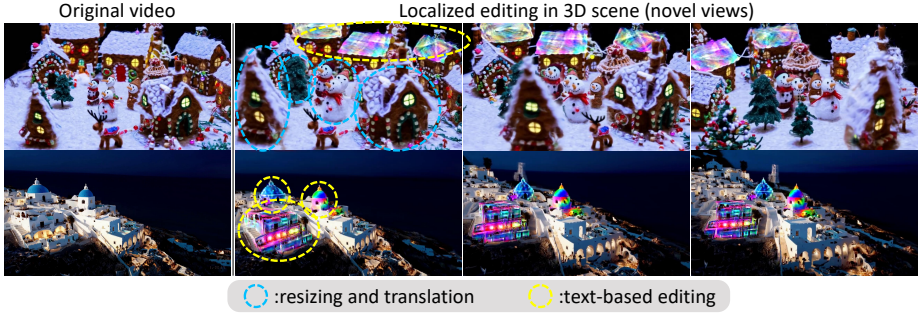

Fig. 4: Versatile applications of Click-Gaussian on synthetic videos generated by Sora. After pre-training 3DGS on Sora-generated videos, users can flexibly modify the reconstructed 3D scenes in real-time, including resizing, translation (sky blue circle), and text-based editing (yellow circle). In the Snow-village scene (first row), we manipulated the scene by enlarging and translating three snowmen, two houses, and a tree, while stylizing other house roofs to crystal. In the Santorini scene (second row), we applied text-based editing to buildings, transforming them into cyberpunk neon, crystal, and rainbow styles from the bottom left, respectively.

[1] https://cdn.openai.com/sora/videos/santorini.mp4

[2] https://cdn.openai.com/tmp/s/interp/b2.mp4. This video has no official name, so we refer to it as Snow-village.

D.2 Open-vocabulary 3D Object Localization

Once trained, our method can perform open-vocabulary 3D object localization as shown in Fig. 5, using the obtained global feature candidates, which we call global clusters. Specifically, for all two-level global clusters, we render only the Gaussians corresponding to each cluster in multiple views (10 randomly sampled views) as shown in Fig. 6. We then input these rendered images into the CLIP image encoder [4] to obtain the CLIP embeddings of each cluster. Thanks to the real-time rendering speed of 3DGS, this process of obtaining CLIP embeddings for all global clusters completes in 20–40 seconds, depending on the number of global clusters in the scene. Note that this process only needs to be performed once before any text query. Subsequently, given text queries, open-vocabulary 3D object localization is performed by returning the global cluster with the highest cosine similarity between the obtained image embeddings of all global clusters and the text query embedding. Fig. 5 qualitatively demonstrates that our approach precisely localizes 3D objects for given text queries using globally obtained clusters.

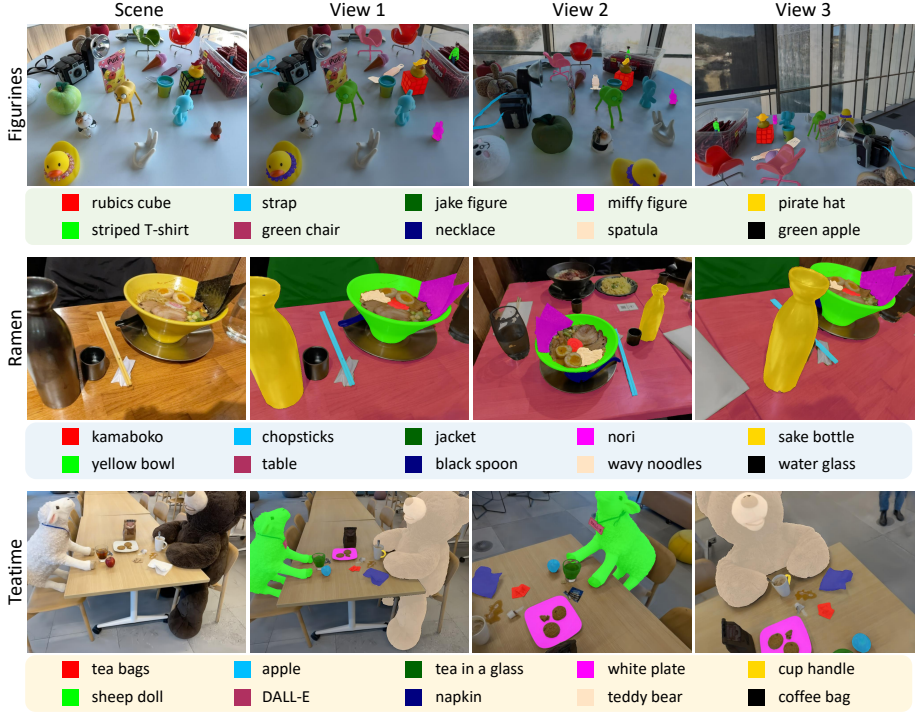

Fig. 5: Open-vocabulary 3D object localization results on the LERF-Mask Dataset. Segmentation results are color-overlaid for visualization in three different scenes.

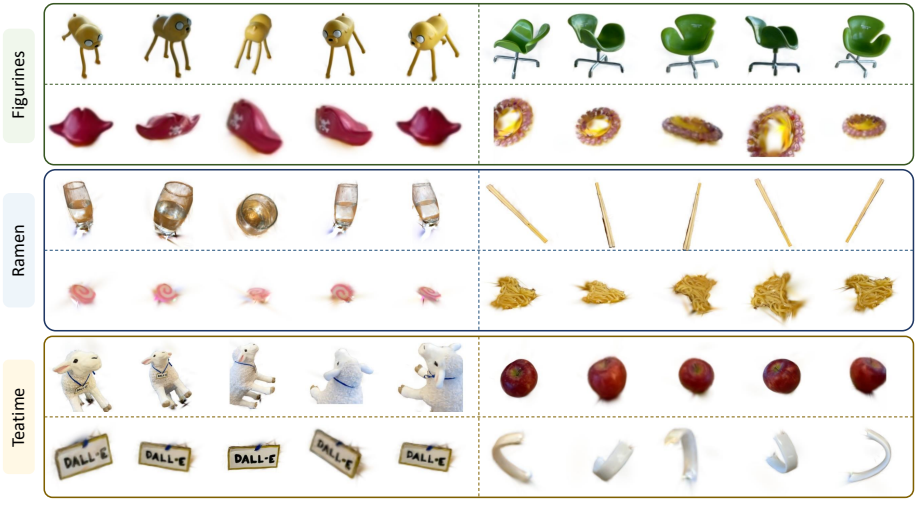

Fig. 6: Examples of rendered images using the Gaussians corresponding to each global cluster. Five representative images are shown per cluster for simplicity. These images are used to obtain CLIP embeddings for each cluster via the CLIP image encoder.

D.3 Additional Results for 3D Segmentation

Experiments on LeRF Dataset. In addition to user-guided segmentation, our approach can also automatically segment everything by calculating the cosine similarity between the rendered 2D feature map and global clusters' features, assigning a global cluster ID with the maximum similarity value to each pixel. By performing this process for each of the two granularity levels, we obtain automatic segmentation results at both coarse and fine levels. Figs. 7, 8, and 9 show the results of automatic segmentation for several complicated real-world scenes from the LeRF dataset [3], along with PCA visualizations of rendered feature maps at two levels. These results qualitatively demonstrate that Click-Gaussian achieves high-fidelity, fine-grained segmentation of everything in complex real-world scenes.

Experiments on SPIn-NeRF Dataset. We further showcase the 3D multi-view segmentation results on the SPIn-NeRF Dataset using the label propagation method, as illustrated in Fig. 10. These results offer additional examples demonstrating the effectiveness of Click-Gaussian across various real-world scenes.

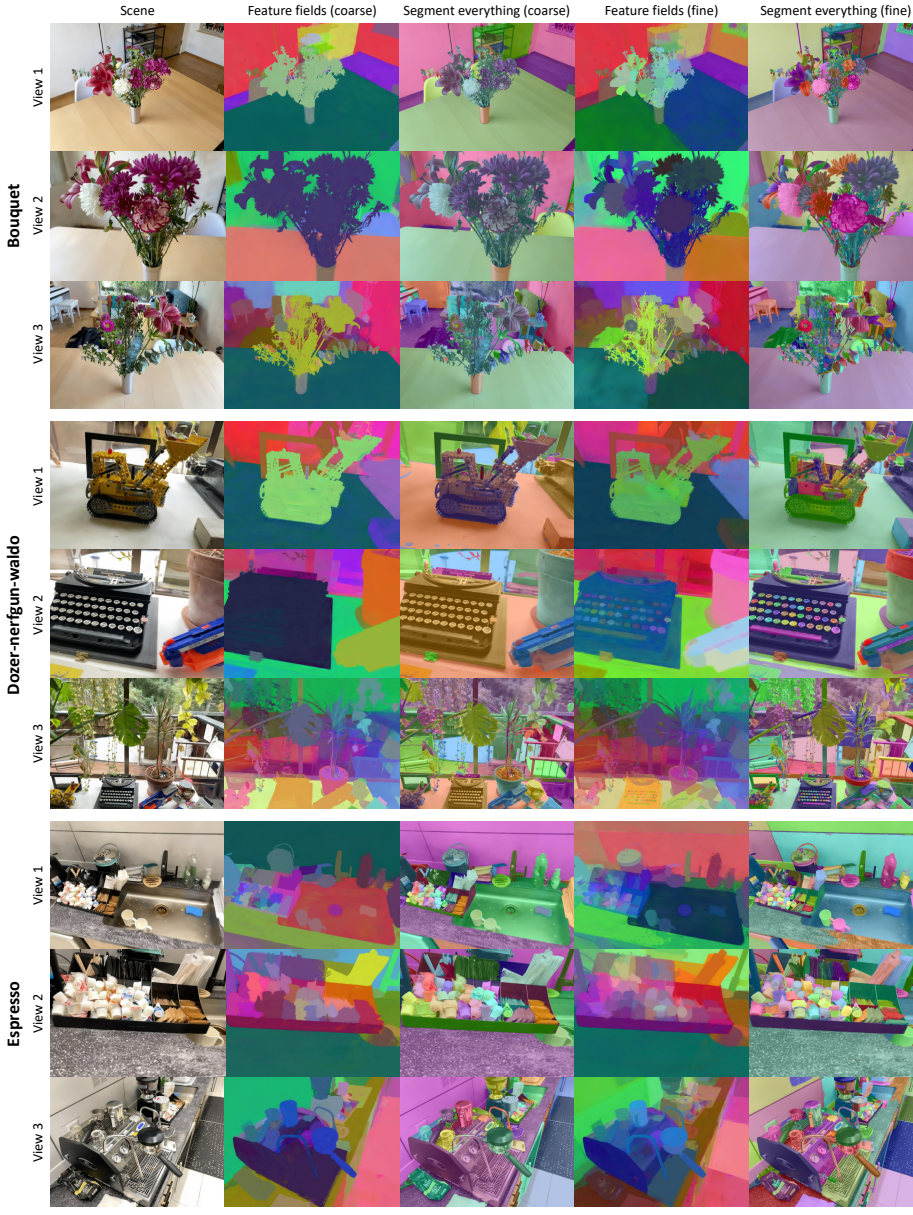

Fig. 7: Segmentation of everything results on the LeRF Dataset. We present automatic segmentation results (third and fifth columns) along with PCA visualizations of rendered feature maps (second and fourth columns) at two granularity levels for Bouquet, Dozer-nerfgun-waldo, and Espresso scenes (first column) from the LeRF Dataset. Objects classified with the same ID in the segmentation results share the same overlaid color across the three given views, as each global cluster ID remains consistent throughout a scene.

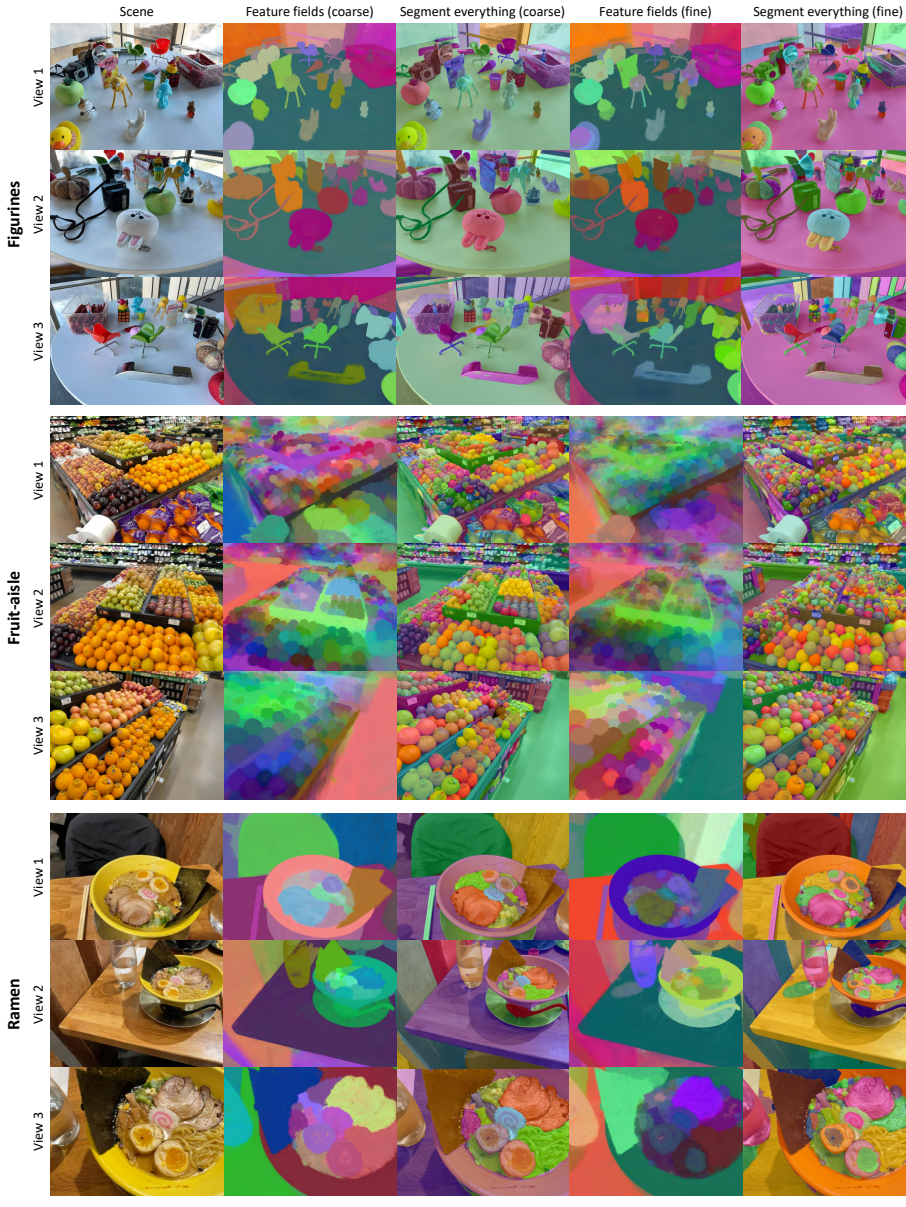

Fig. 8: Segmentation of everything results on the LeRF Dataset. We present automatic segmentation results (third and fifth columns) along with PCA visualizations of rendered feature maps (second and fourth columns) at two granularity levels for Figurines, Fruit-aisle, and Ramen scenes (first column) from the LeRF Dataset. Objects classified with the same ID in the segmentation results share the same overlaid color across the three given views, as each global cluster ID remains consistent throughout a scene.

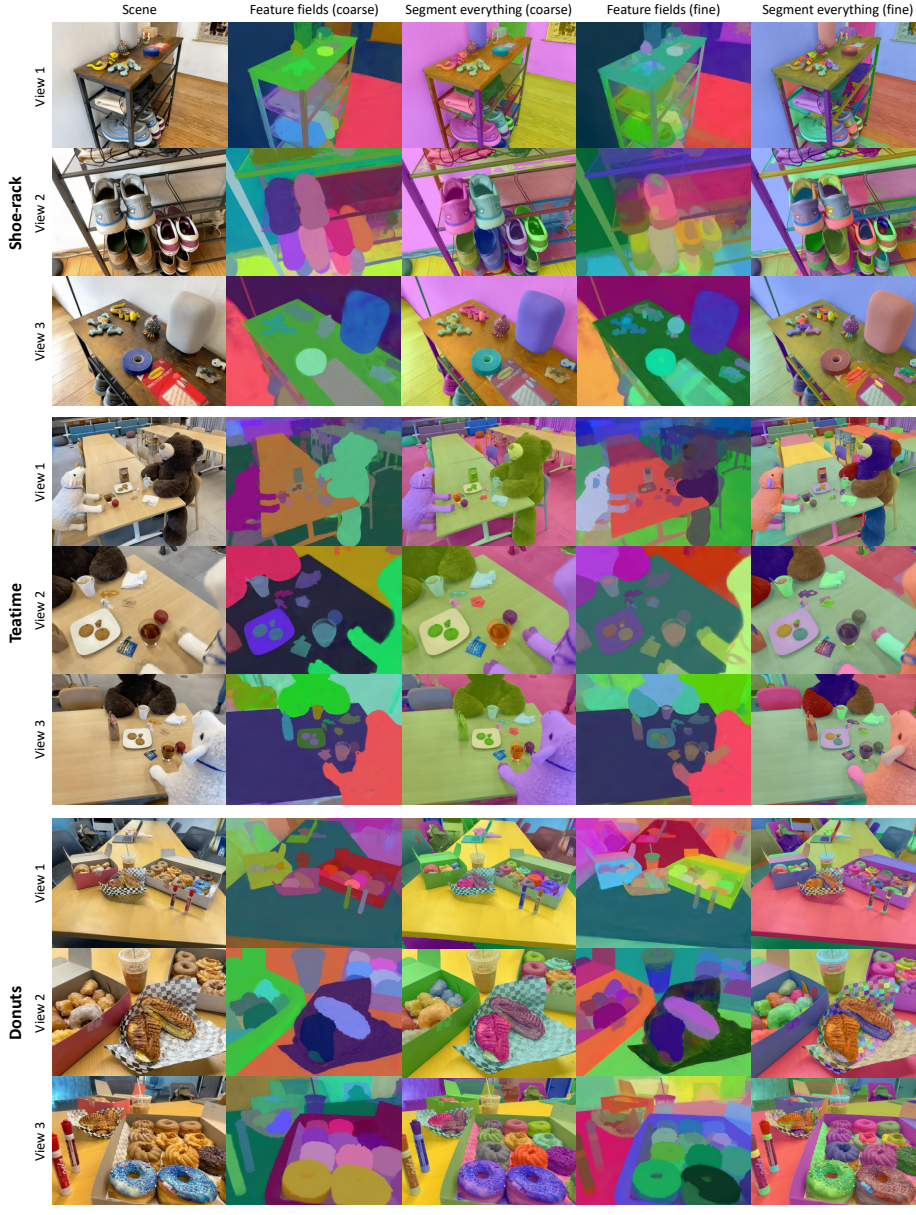

Fig. 9: Segmentation of everything results on the LeRF Dataset. We present automatic segmentation results (third and fifth columns) along with PCA visualizations of rendered feature maps (second and fourth columns) at two granularity levels for Shoe-rack, Teatime, and Donuts scenes (first column) from the LeRF Dataset. Objects classified with the same ID in the segmentation results share the same overlaid color across the three given views, as each global cluster ID remains consistent throughout a scene.

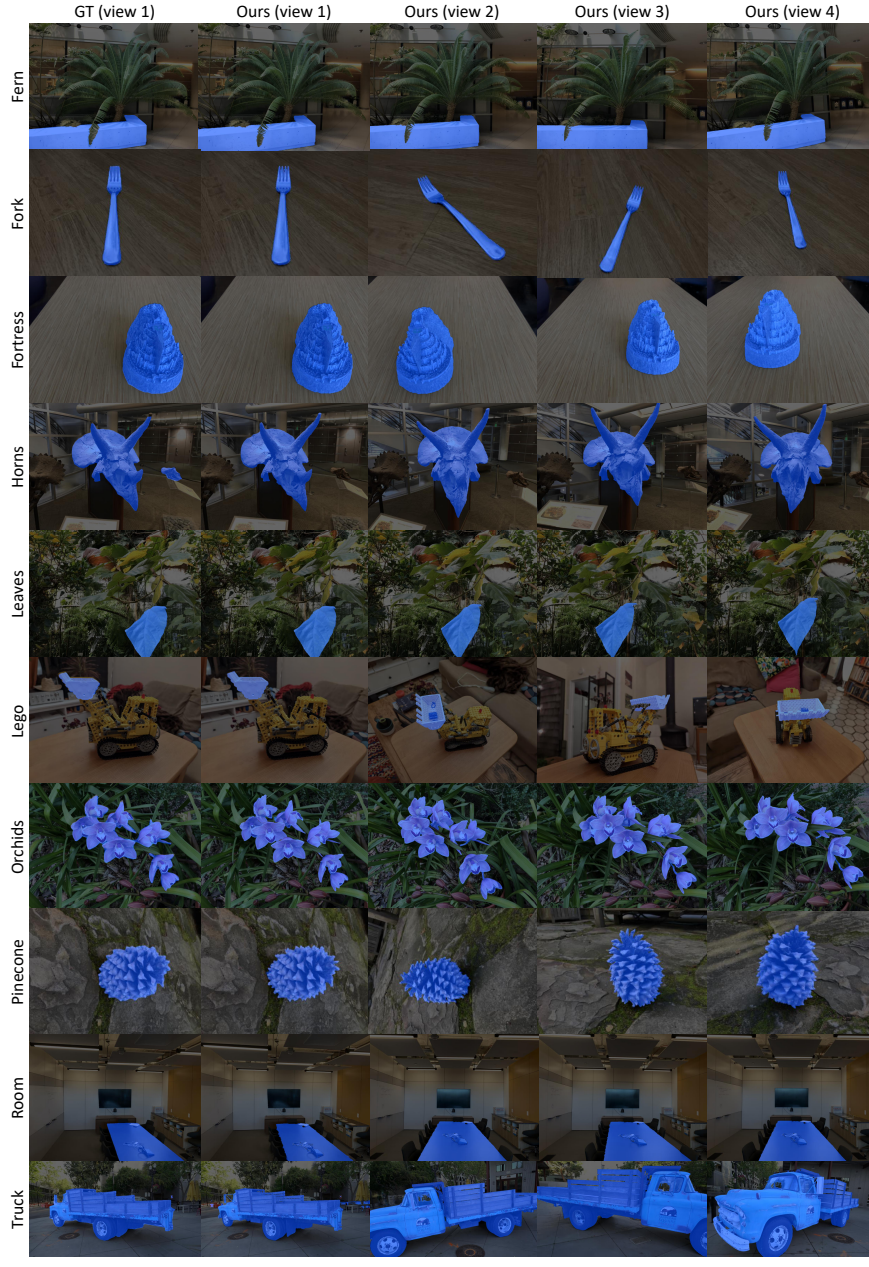

Fig. 10: 3D segmentation results on the SPIn-NeRF Dataset. We use the label propagation method based on the ground truth mask of a reference view (first column) to identify the cluster IDs belonging to the target object. These IDs are then used to generate 2D masks for test views (subsequent columns).

References

1. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024), https://openai.com/research/video-generation-models-as-world-simulators
2. Hoffstadt, J., Cothren, P., Contributors: Dearpygui. https://github.com/hoffstadt/DearPyGui
3. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19729–19739 (2023)
4. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
5. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
6. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision (ECCV) (2016)
7. Skalski, P.: Make Sense. https://github.com/SkalskiP/make-sense/ (2019)
8. Tang, J., Chen, X., Wan, D., Wang, J., Zeng, G.: Segment-anything nerf. https://github.com/ashawkey/Segment-Anything-NeRF (2023)
9. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. arXiv preprint arXiv:2312.00732 (2023)